

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-037843

(43)Date of publication of application : 07.02.2003

(51)Int.Cl.

H04N 7/30
H03M 7/30
H04N 5/92
H04N 7/32

(21)Application number : 2001-221674

(71)Applicant : SONY CORP

(22)Date of filing : 23.07.2001

(72)Inventor : TAKAHASHI KUNIAKI
SATO KAZUFUMI
YAGASAKI YOICHI

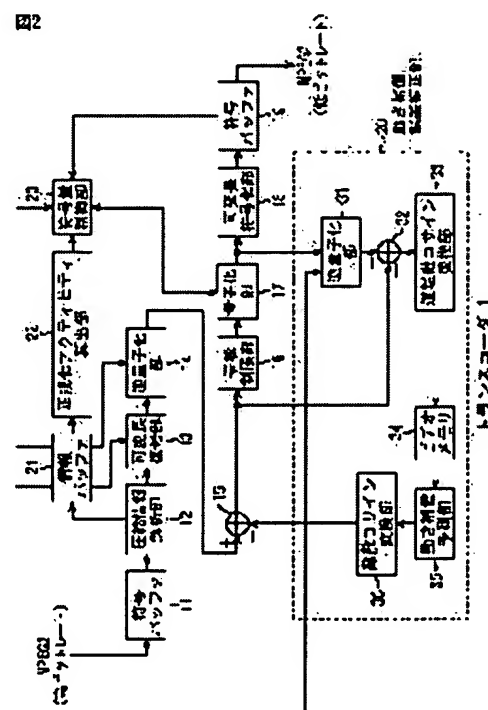
(54) PICTURE PROCESSOR, METHOD THEREFOR, RECORDING MEDIUM AND PROGRAM THEREOF

(57)Abstract:

PROBLEM TO BE SOLVED: To realize a process of changing the bit rate of compressed picture information at a high speed.

SOLUTION: A normalized activity calculator 22 calculates a normalized activity from luminance component pixel values of an original picture, based on the bit rate, the bit quantity generated per frame, the quantizing step size per microblock, and generated bit quantity information from an information buffer 21; and outputs the activity to a code quantity controller 23. The controller 23 calculates a quantizing scale code of a target bit rate, corresponding to visual characteristics by utilizing the normalized activity inputted from the calculator 22, and outputs the code to a quantizer 17.

The quantizer 17 quantizes discrete cosine transform coefficients inputted from a band limiter 16, based on the inputted quantizing scale code.



LEGAL STATUS

[Date of request for examination] 20.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-37843

(P2003-37843A)

(43) 公開日 平成15年2月7日(2003.2.7)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード(参考)		
H 0 4 N	7/30	H 0 3 M	7/30	A	5 C 0 5 3
H 0 3 M	7/30	H 0 4 N	7/133	Z	5 C 0 5 9
H 0 4 N	5/92		7/137	Z	5 J 0 6 4
	7/32		5/92	H	

審査請求 未請求 請求項の数10 O L (全 17 頁)

(21) 出願番号 特願2001-221674(P2001-221674)

(22) 出願日 平成13年7月23日(2001.7.23)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 高橋 邦明

東京都品川区北品川6丁目7番35号 ソニ

一株式会社内

(72) 発明者 佐藤 数史

東京都品川区北品川6丁目7番35号 ソニ

一株式会社内

(74) 代理人 100082131

弁理士 稲本 義雄

最終頁に続く

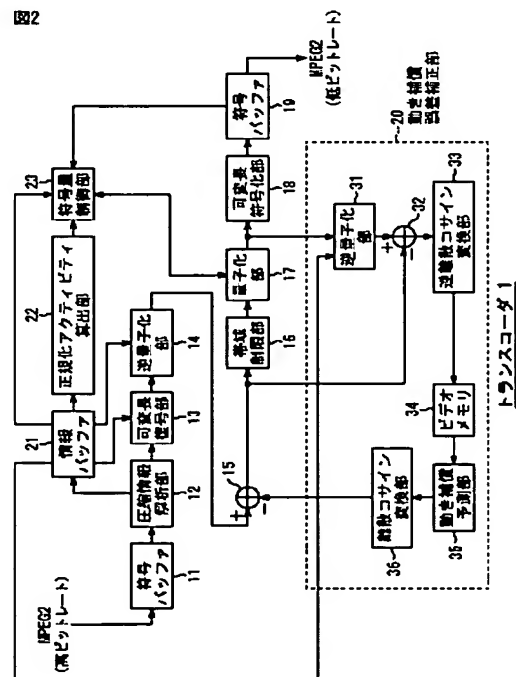
(54) 【発明の名称】 画像処理装置および方法、記録媒体、並びにプログラム

(57) 【要約】

【課題】 画像圧縮情報のビットレートの変換処理を高速で実現する。

【解決手段】 正規化アクティビティ算出部22は、情報バッファ21よりビットレート、フレーム毎の発生ビット量、及びマクロブロック毎の量子化ステップサイズ、並びに発生ビット量情報に基づいて、原画の輝度成分画素値より算出される正規化アクティビティを算出し、符号量制御部23に出力する。符号量制御部23は、目標とするビットレートの、視覚特性に対応した量子化スケールコードを、正規化アクティビティ算出部22より入力された正規化アクティビティを利用して算出し、量子化部17に出力する。量子化部17は、入力された量子化スケールコードに基づいて、帯域制限部16より入力された離散コサイン変換係数を量子化する。

図2



【特許請求の範囲】

【請求項1】 第1のビットレートで符号化された画像データを第2のビットレートで符号化された画像データに変換する画像処理装置において、

前記第1のビットレートで符号化された画像データに基づいて、前記画像データを前記第1のビットレートで符号化するために必要な第1の量子化スケールコードを演算する第1の量子化スケールコード演算手段と、

前記第1のビットレートで符号化された画像データに基づいて、前記第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算する第2の量子化スケールコード演算手段と、

前記第1の量子化スケールコードと前記第2の量子化スケールコードに基づいて、正規化アクティビティを演算する正規化アクティビティ演算手段と、

前記画像データを前記第2のビットレートで符号化するために必要とされる第3の量子化スケールコードを演算する第3の量子化スケールコード演算手段と、

前記第3の量子化スケールコードと前記正規化アクティビティに基づいて、前記第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードを演算する第4の量子化スケールコード演算手段とを備えることを特徴とする画像処理装置。

【請求項2】 前記第2のビットレートは、前記第1のビットレートよりも低ビットレートであることを特徴とする請求項1に記載の画像処理装置。

【請求項3】 前記符号化された画像データは、MPEG方式で圧縮された画像データであることを特徴とする請求項1に記載の画像処理装置。

【請求項4】 前記第1のビットレートで符号化された画像データは、それ自身の符号量、フレーム毎の発生ビット量、マクロブロック毎の発生ビット量、または、マクロブロック毎の量子化ステップサイズを含むことを特徴とする請求項1に記載の画像処理装置。

【請求項5】 前記第1の量子化スケールコード演算手段は、前記符号量、前記フレーム毎の発生ビット量、および、前記マクロブロック毎の発生ビット量に基づいて、前記第1の量子化スケールコードを演算することを特徴とする請求項4に記載の画像処理装置。

【請求項6】 前記第2の量子化スケールコード演算手段は、前記マクロブロック毎の量子化ステップサイズを2で除することにより、前記第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算することを特徴とする請求項4に記載の画像処理装置。

【請求項7】 前記正規化アクティビティ演算手段は、前記第1の量子化スケールコードで前記第2の量子化スケールコードを除することにより前記正規化アクティビティを演算することを特徴とする請求項1に記載の画像処理装置。

【請求項8】 第1のビットレートで符号化された画像データを第2のビットレートで符号化された画像データに変換する画像処理装置の画像処理方法において、

前記第1のビットレートで符号化された画像データに基づいて、前記画像データを前記第1のビットレートで符号化するために必要な第1の量子化スケールコードを演算する第1の量子化スケールコード演算ステップと、

前記第1のビットレートで符号化された画像データに基づいて、前記第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算する第2の量子化スケールコード演算ステップと、

前記第1の量子化スケールコードと前記第2の量子化スケールコードに基づいて、正規化アクティビティを演算する正規化アクティビティ演算ステップと、

前記画像データを前記第2のビットレートで符号化するために必要とされる第3の量子化スケールコードを演算する第3の量子化スケールコード演算ステップと、

前記第3の量子化スケールコードと前記正規化アクティビティに基づいて、前記第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードを演算する第4の量子化スケールコード演算ステップとを含むことを特徴とする画像処理方法。

【請求項9】 第1のビットレートで符号化された画像データを第2のビットレートで符号化された画像データに変換する画像処理装置を制御するプログラムであって、

前記第1のビットレートで符号化された画像データに基づいて、前記画像データを前記第1のビットレートで符号化するために必要な第1の量子化スケールコードの演算を制御する第1の量子化スケールコード演算制御ステップと、

前記第1のビットレートで符号化された画像データに基づいて、前記第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードの演算を制御する第2の量子化スケールコード演算制御ステップと、

前記第1の量子化スケールコードと前記第2の量子化スケールコードに基づいて、正規化アクティビティの演算を制御する正規化アクティビティ演算制御ステップと、

前記画像データを前記第2のビットレートで符号化するために必要とされる第3の量子化スケールコードの演算を制御する第3の量子化スケールコード演算制御ステップと、

前記第3の量子化スケールコードと前記正規化アクティビティに基づいて、前記第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードの演算を制御する第4の量子化スケールコード演算制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項10】 第1のビットレートで符号化された画像データを第2のビットレートで符号化された画像データに変換する画像処理装置を制御するコンピュータに、前記第1のビットレートで符号化された画像データに基づいて、前記画像データを前記第1のビットレートで符号化するために必要な第1の量子化スケールコードの演算を制御する第1の量子化スケールコード演算制御ステップと、

前記第1のビットレートで符号化された画像データに基づいて、前記第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードの演算を制御する第2の量子化スケールコード演算制御ステップと、

前記第1の量子化スケールコードと前記第2の量子化スケールコードに基づいて、正規化アクティビティの演算を制御する正規化アクティビティ演算制御ステップと、前記画像データを前記第2のビットレートで符号化するために必要とされる第3の量子化スケールコードの演算を制御する第3の量子化スケールコード演算制御ステップと、

前記第3の量子化スケールコードと前記正規化アクティビティに基づいて、前記第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードの演算を制御する第4の量子化スケールコード演算制御ステップとを実行させるプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、画像処理装置および方法、記録媒体、並びにプログラムに関し、特に、MP EG (Moving Picture Experts Group) などの離散コサイン変換等の直交変換と動き補償によって圧縮された画像情報（ビットストリーム）を、より低ビットレートに変換する処理を、高速に実現できるようにした画像処理装置および方法、記録媒体、並びにプログラムに関する。

【0002】

【従来の技術】近年、画像データをデジタル情報として取り扱い、効率の高い情報の伝送、蓄積を目的とするため、画像データ特有の冗長性を利用した離散コサイン変換等の直交変換と動き補償により圧縮するMPEGなどの方式に準拠した装置が、放送局などの情報配信用として、または、一般家庭における情報受信用として普及しつつある。

【0003】特に、MPEG 2 (ISO/IEC 13818-2) は、汎用画像符号化方式として定義されたものであり、飛び越し走査画像、および、順次走査画像、並びに、標準解像度画像、および、高精細画像に対応する標準規格である。このためMPEG 2は、放送業界などの業務用途、および、一般ユーザ向けの用途と言った広範なアプリケーションソフトウェアに今後とも用いられるものと予想される。また、MPEG 2の圧縮方式を用いることにより、例え

ば、720画素×480画素から構成される標準解像度の飛び越し走査画像であれば4乃至8 Mbps、また、1920画素×1088画素から構成される高解像度の飛び越し走査画像であれば18乃至22 Mbpsの符号量（ビットレート）を、それぞれに割り当てることで、高い圧縮率と良好な画質を維持しつつ圧縮することができ、その伝送や蓄積を実現することができる。

【0004】しかしながら、高解像度の画像情報は膨大であり、上記で述べた様なMPEG等の符号化方式を用いて圧縮しても、十分な画質を得るためには、例えば、1920画素×1080画素の30Hzの飛び越し走査画像の場合、18乃至22 Mbps程度、または、それ以上の符号量（ビットレート）を必要とするといった問題があった。すなわち、例えば、衛星放送や、ケーブルテレビジョンなどのネットワークを介してMPEG方式で圧縮された画像情報を伝送する場合、伝送路の帯域幅に合わせて、また、光ディスク、磁気ディスク、若しくは、光磁気ディスクのような記録媒体上でMPEG方式で圧縮された画像情報を蓄積（記録）する場合、記録媒体の容量に合わせて、画質劣化を最小限に抑えながら更なる符号量（ビットレート）の削減を行う必要がある。このような符号量（ビットレート）の削減の必要性は、高解像度のみならず、標準解像度画像（例えば、720画素×480画素の30Hzの飛び越し走査画像）の画像圧縮情報（ビットストリーム）を、上述の様に、ネットワークを介して伝送する場合や、記録媒体上に記録させる場合にも生じることが考えられる。

【0005】このような問題を解決する手段として、階層符号化（スケーラビリティ）処理や画像情報変換（トランスコーディング）処理といった手法が考えられている。前者に関連して、MPEG 2ではSNR (Signal to Noise Ratio) スケーラビリティが標準化されており、これによって、高SNRの画像圧縮情報（ビットストリーム）と低SNRの画像圧縮情報（ビットストリーム）を階層的に符号化することを可能にしている。しかしながら、階層符号化を行う為には、符号化の時点で、ネットワーク媒体の帯域幅、または、記録媒体の記憶容量の拘束条件が既知である必要があるが、実際のシステムにおいて、これらの情報は未知であることが殆どである。従って、後者の方が、実際のシステムに則した、より自由度の高い方式であると言える。

【0006】この画像情報変換（トランスコーディング）処理とは、例えば、MPEG 2方式で圧縮された画像圧縮情報を、より低ビットレートの画像圧縮情報に変換する処理である。この画像情報変換処理では、まず、MPEG 2方式で圧縮された画像圧縮情報からピクチャコーディングタイプ、各マクロブロック毎の量子化幅、および、量子化行列と言った情報が抽出される。その後、画像圧縮情報は、可変長復号され、量子化された離散コサイン変換係数として2次元データに再配列される。続いて、量子化された2次元データに再配列された離散コサイン

変換係数は、上述の量子化幅、および、量子化行列に基づいて、逆量子化される。さらに、逆量子化された離散コサイン変換係数は、所定の高域成分係数が削減されて、ターゲットとなる（元のビットレートより低い）ビットレートに基づいて生成される量子化幅（量子化スケールコード）で再量子化された後、再びMPEG2方式で可変長符号化されて、出力される。

【0007】ここで、MPEG2方式の画像圧縮情報に対応した量子化幅（量子化スケールコード）は、図1のフローチャートで説明する処理により、符号量制御が行われる。尚、以下の説明においては、MPEG2 Test Model 5（ISO/IEC JTC1/SC19/WG11N400）方式で圧縮された画像圧縮情報を例として説明する。また、この符号量制御では、目標符号量（ターゲットビットレート）、及び、GOP（Group of Pictures）構成が入力変数であるものとする。ここでいうGOPとは、MPEG2方式で画像圧縮される際に、使用されるI（Intra Code）ピクチャ（それ自身で単独に符号化されるピクチャ）、P（Predictive Code）ピクチャ（時間的に前方の（過去の）Iピクチャ、または、Pピクチャにより符号化されるピクチャ）、および、B（Bidirectionally Predictive Code）ピクチャ（時間的に前後の（過去、または、未来の）Iピクチャ、または、Pピクチャにより符号化されるピクチャ）の3種類のピクチャタイプがグループ化されたものである。

【0008】ステップS1において、GOP内の各ピクチャに対する割当ビット量が、割り当て対象ピクチャを含めGOP内でまだ復号されていないピクチャに対して割り当てられるビット量（以下、これを割り当てビット量Rとする）に基づいて配分される。この配分がGOP内の符号化ピクチャ順に繰り返される。その際、以下に述べる二つの仮定を用いて各ピクチャへの符号量の割当を行う。

【0009】すなわち、第1の仮定として、各ピクチャを符号化する際に用いる平均量子化スケールコードと発生符号量の積は、画面が変化しない限り、ピクチャタイプ毎に、一定になると仮定する。そこで、画面の複雑さを示す変数 X_i 、 X_p 、 X_b (global complexity measure) は、以下の式(1)乃至式(3)により更新される。このパラメータにより次のピクチャを符号化する際の量子化スケールコードと発生符号量との関係が推定される。

【0010】 $X_i = S_i Q_i \cdots (1)$
 $X_p = S_p Q_p \cdots (2)$
 $X_b = S_b Q_b \cdots (3)$

【0011】ここで S_i 、 S_p 、 S_b はピクチャ符号化時の発生符号ビット量であり、 Q_i 、 Q_p 、 Q_b は、ピクチャ符号化時の平均量子化スケールコードである。また、初期値は、目標符号量（ターゲットビットレート） bit_rate (bits/sec)を用いて、以下の式(4)乃至式(6)のように示される値とする。

【0012】 $X_i = 160 \times bit_rate / 115 \cdots (4)$
 $X_p = 60 \times bit_rate / 115 \cdots (5)$
 $X_b = 42 \times bit_rate / 115 \cdots (6)$

【0013】第2の仮定として、Iピクチャの量子化スケールコードを基準としたP、Bピクチャの量子化スケールコードの比率 K_p 、 K_b が式(7)、式(8)で定められた値となる場合、常に全体の画質が最適化されると仮定する。

【0014】 $K_p = Q_p / Q_i \cdots (7)$
 $K_b = Q_b / Q_i \cdots (8)$

【0015】すなわち、Bピクチャの量子化スケールコードは、I、Pピクチャの量子化スケールコードの常に1.4倍としている。これは、BピクチャがI、Pピクチャに比較して多少粗めに符号化されることにより、Bピクチャで節約できる符号量をI、Pピクチャに加えると、I、Pピクチャの画質が改善され、これを参照するBピクチャの画質も改善されることを想定している。

【0016】上記二つの仮定により、GOPの各ピクチャに対する割当ビット量(T_i 、 T_p 、 T_b)は式(9)乃至式(11)で示す値であるとする。

【0017】

$$T_i = \max \{ R / (1 + N_p X_p / X_i K_p + N_b X_b / X_i K_b), bit_rate / (8 \times picture_rate) \} \cdots (9)$$

$$T_p = \max \{ R / (N_p + N_b K_p X_b / X_b K_p), bit_rate / (8 \times picture_rate) \} \cdots (10)$$

$$T_b = \max \{ R / (N_b + N_p K_b X_p / X_p K_b), bit_rate / (8 \times picture_rate) \} \cdots (11)$$

ここで N_p 、 N_b はGOP内でまだ符号化されていないP、Bピクチャの枚数である。このようにして求めた割当符号量を基にして、各ピクチャを符号化する毎に、GOP内の未符号化ピクチャに対して割り当てられる割り当てビット量Rを、以下の式(12)により更新する。

$$R = R - S_{i,p,b} \cdots (12)$$

【0019】また、GOPの最初のピクチャを符号化する際には、式(13)により割り当てビット量Rを更新する。

【0020】

$$R = bit_rate \times N / picture_rate + R \cdots (13)$$

【0021】ここで、NはGOP内のピクチャ数である。また、シーケンスの最初の割り当てビット量Rの初期値は0とする。

【0022】ステップS2において、ステップS1の処理で式(9)乃至式(11)により求められた各ピクチャに対する割当ビット量(T_i 、 T_p 、 T_b)と、実際の発生符号量を一致させるため、各ピクチャ毎に独立に設定した3種類の仮想バッファの容量を基に、量子化スケールコードが、マクロブロック単位のフィードバック制御で求

められる。尚、以下の説明において、マクロブロックは 8×8 の2次元の構成であるものとする。

【0023】すなわち、 j 番目のマクロブロックの符号化に先立ち、仮想バッファの占有量を式(14)乃至式(16)によって求める。

【0024】

$$dji = d0i + Bj - 1 - Ti \times (j - 1) / MB\ cnt \cdots (14)$$

$$djp = d0p + Bj - 1 - Tp \times (j - 1) / MB\ cnt \cdots (15)$$

$$djb = d0b + Bj - 1 - Tb \times (j - 1) / MB\ cnt \cdots (16)$$

【0025】ここで、 $d0i$ 、 $d0p$ 、 $d0b$ は、 i ピクチャ、 P ピクチャ、および、 B ピクチャのそれぞれの仮想バッファの初期占有量を示し、 Bj はピクチャの先頭から j 番目のマクロブロックまでの発生ビット量、 $MBcnt$ は1ピクチャ内のマクロブロック数である。

【0026】各ピクチャ符号化終了時の仮想バッファ占有量 ($dMBcnti$ 、 $dMBcntp$ 、 $dMBcntb$) はそれぞれ同一ピクチャタイプで、次のピクチャに対する仮想バッファ占有量の初期値 ($d0i$ 、 $d0p$ 、 $d0b$) として用いられる。

【0027】次に、 j 番目のマクロブロックに対する量子化スケールコードを、以下のように示す式(17)により計算する。

$$Qj = dj \times 31 / r \cdots (17)$$

【0029】ここで、 r はリアクションパラメータと呼ばれるフィードバックループの応答速度を制御するパラ

$$act_j = 1 + \min_{sblk=1,8} (var_sblk)$$

【数2】

$$var_sblk = \frac{1}{64} \sum_{k=1}^{64} (P_k - \bar{P})^2 \cdots (23)$$

【数3】

$$\bar{P} = \frac{1}{64} \sum_{k=1}^{64} P_k \cdots (24)$$

【0036】ここで P_k は原画の輝度信号ブロック内画素値である。式(22)において最小値をとるのは、マクロブロック内の一部だけでも平坦部のある場合には量子化を細かくするためである。

$$Nactj = (2 \times actj + avg\ act) / (actj + 2 \times avg\ act) \cdots (25)$$

【0039】ここで、 Avg_act は、直前に符号化したピクチャでの $actj$ の平均値である。視覚特性を考慮した、量子化スケールコード $mquantj$ はステップ2で得られた量子化スケールコード Qj の値に基づいて式(26)で与えられる。

$$mquantj = Qj \times Nactj \cdots (26)$$

【0041】以上のように求められた量子化スケールコード $mquantj$ により、MPEG2方式で圧縮された画像圧縮情報が、目標とする、より低ビットレート(ターゲットビットレート)の画像圧縮情報に変換される。

メータであり、以下のように示す式(18)で与えられる。

【0030】

$$r = 2 \times \text{bit rate} / \text{picture rate} \cdots (18)$$

【0031】尚、シーケンスの最初における仮想バッファの初期値は、以下のように示す式(19)乃至式(21)で与えられる。

$$d0i = 10 \times r / 31 \cdots (19)$$

$$d0p = Kp \cdot d0i \cdots (20)$$

$$d0b = Kb \cdot d0i \cdots (21)$$

【0033】ステップS3において、ステップS2の処理で求められた量子化スケールコードが、視覚的に劣化の目立ちやすい平坦部で、より細かく量子化され、劣化の目立ちにくい絵柄の複雑な部分で粗く量子化されるように、各マクロブロック毎のアクティビティと呼ばれる変数によって変化させられる。

【0034】アクティビティは、原画の輝度信号画素値を用い、フレーム離散コサイン変換モードにおける4個のブロックと、フィールド離散コサイン変換モードにおける4個のブロックとの、合計8ブロックの画素値を用いて、以下のように示す式(22)乃至式(24)で与えられる。

【0035】

【数1】

$$\cdots (22)$$

【0037】更に、式(25)により、その値が0.5乃至2の範囲を取る正規化アクティビティ $Nactj$ を求める。

【0038】

【0042】

【発明が解決しようとする課題】しかしながら、上記のような手法では、画像変換処理を実行する度に、式(22)乃至式(24)において、各マクロブロックに対する画素値の平均値を全て計算する必要があり、その演算にかかる処理量が膨大なものになってしまうため、結果として処理時間がかかってしまううえ、これらの膨大な演算に対応できるハードウェアを必要することから、装置コストが増大してしまうという課題があった。

【0043】また、上述のアクティビティは、原画像の

輝度信号画素値を用いて計算されているが、画像変換処理において原画像の輝度信号画素値を知ることはできないため、入力となる画像圧縮情報が、例えば肌色検出、または、赤色検出により、画像の複雑度に合わせた効率的な適応量子化が行われていると、再量子化時に同様の正規化アクティビティ情報を用いた適応量子化が行えないという課題があった。

【0044】本発明はこのような状況に鑑みてなされたものであり、MPEGなどの離散コサイン変換等の直交変換と動き補償によって圧縮され、符号化された画像情報（ビットストリーム）が、衛星放送、ケーブルテレビジョン、または、インターネットなどのネットワークを介して伝送される場合や、光ディスク、磁気ディスク、または、光磁気ディスクのような記憶媒体に記録（蓄積）される場合に、より低ビットレートの画像データに変換される処理を、高速に実現できるようにするものである。

【0045】

【課題を解決するための手段】本発明の画像処理装置は、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードを演算する第1の量子化スケールコード演算手段と、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算する第2の量子化スケールコード演算手段と、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティを演算する正規化アクティビティ演算手段と、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードを演算する第3の量子化スケールコード演算手段と、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードを演算する第4の量子化スケールコード演算手段とを備えることを特徴とする。

【0046】前記第2のビットレートは、第1のビットレートよりも低ビットレートとするようにすることができる。

【0047】前記符号化された画像データは、MPEG方式で圧縮された画像データとするようにすることができる。

【0048】前記第1のビットレートで符号化された画像データには、それ自身の符号量、フレーム毎の発生ビット量、マクロブロック毎の発生ビット量、または、マクロブロック毎の量子化ステップサイズを含ませるようにすることができる。

【0049】前記第1の量子化スケールコード演算手段には、符号量、フレーム毎の発生ビット量、および、マ

クロブロック毎の発生ビット量に基づいて、第1の量子化スケールコードを演算させるようにすることができる。

【0050】前記第2の量子化スケールコード演算手段には、マクロブロック毎の量子化ステップサイズを2で除することにより、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算させるようにすることができる。

【0051】前記正規化アクティビティ演算手段には、第1の量子化スケールコードで第2の量子化スケールコードを除することにより正規化アクティビティを演算させるようにすることができる。

【0052】本発明の画像処理方法は、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードを演算する第1の量子化スケールコード演算ステップと、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算する第2の量子化スケールコード演算ステップと、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティを演算する正規化アクティビティ演算ステップと、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードを演算する第3の量子化スケールコード演算ステップと、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードを演算する第4の量子化スケールコード演算ステップとを含むことを特徴とする。

【0053】本発明の記録媒体のプログラムは、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードの演算を制御する第1の量子化スケールコード演算制御ステップと、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードの演算を制御する第2の量子化スケールコード演算制御ステップと、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティの演算を制御する正規化アクティビティ演算制御ステップと、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードの演算を制御する第3の量子化スケールコード演算制御ステップと、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードの演算を制御する第4の量子化スケールコード演算制御ステップとを含むことを

特徴とする。

【0054】本発明のプログラムは、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードの演算を制御する第1の量子化スケールコード演算制御ステップと、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードの演算を制御する第2の量子化スケールコード演算制御ステップと、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティの演算を制御する正規化アクティビティ演算制御ステップと、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードの演算を制御する第3の量子化スケールコード演算制御ステップと、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードの演算を制御する第4の量子化スケールコード演算制御ステップとをコンピュータに実行させることを特徴とする。

【0055】本発明の画像処理装置および方法、並びにプログラムにおいては、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードが演算され、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードが演算され、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティが演算され、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードが演算され、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードが演算される。

【0056】

【発明の実施の形態】図2は、本発明に係るトランスコーダ1の一実施の形態の構成を示す図である。トランスコーダ1は、入力される所定の圧縮方式で圧縮された画像圧縮情報のビットレートを変換して、出力するものである。すなわち、トランスコーダ1は、例えば、既に、MPEG2方式で圧縮されたビットレートが20Mbpsの画像圧縮情報を、同じMPEG2方式で圧縮されたビットレートが16Mbpsの画像圧縮情報に変換するといった処理を行うものである。当然のことながら、入力される画像圧縮情報のビットレートと、変換後の画像圧縮情報のビットレートは、上述のビットレート以外であってもよい。

【0057】符号バッファ11は、入力された高ビットレートの画像圧縮情報（多くの符号量を持つ画像圧縮情

報（ビットストリーム））を格納し、一時的に記憶した後、順次圧縮情報解析部12に出力する。符号バッファ11に格納される画像圧縮情報（ビットストリーム）は、MPEG2で規定されている、図示せぬVBV（Video Buffering Verifier：ビットストリームのオーバーフロー、または、アンダーフローを監視するための装置）の拘束条件を満たすよう符号化されているので、符号バッファ11においては、オーバーフロー、または、アンダーフローが生じないように符号化されている。

【0058】圧縮情報解析部12は、符号バッファ11より入力された画像圧縮情報を、MPEG2で定められた構文（シンタクス）に従って解析し、画像圧縮情報（ビットストリーム）中から、符号量、フレーム毎の発生ビット量、マクロブロック毎の発生ビット量、量子化スケールコード、マクロブロック毎の量子化ステップサイズ、q_scale_typeフラグ、量子化行列、および、走査方式といった情報を抽出し、情報バッファ21に格納させると共に、画像圧縮情報に加えて、量子化幅、量子化行列、および、走査方式の情報を可変長復号部13に出力する。ここで、符号量とは、いわゆるビットレートを示す値である。フレーム毎の発生ビット量、または、マクロブロック毎の発生ビット量とは、圧縮された画像のフレーム単位、または、マクロブロック単位で使用されるビット量である。量子化スケールコードとは、量子化処理を行う際の基準となる値を特定するためのコードである。マクロブロック毎の量子化ステップサイズとは、量子化スケールコードにより特定される量子化量のステップ間隔を示す値である。q_scale_typeフラグとは、量子化する際の量子化スケールコードと量子化値の関係が線形のもの（Linear Q）であるか、または、非線形のもの（Non-Linear Q）であるかを、1、または、0で示すフラグである。量子化行列とは、量子化する際の演算に使用する行列である。走査方式とは、例えば、ジグザグスキャンであるか、または、オルトネートスキャンであるかなどの走査方式を示す情報である。

【0059】可変長復号部13は、まず、圧縮情報解析部12より入力された画像圧縮情報のうち、イントラマクロブロックの直流成分に関しては、隣接するブロックとの差分値として符号化されているデータを、その他の係数に関しては、ラン（符号中のゼロが連続している数）とレベル（符号中のゼロ以外の値）により符号化されたデータを可変長復号し、量子化された一次元の離散コサイン変換係数を得る。次に、可変長復号部13は、圧縮情報解析部12により抽出された、画像の走査方式（ジグザグスキャン、または、オルトネートスキャン）に関する情報に基づいて、量子化された離散コサイン変換係数を二次元データとして再配列し、量子化幅、および、量子化行列の情報と共に逆量子化部14に出力する。尚、走査方式については、図5を参照して後述する。

【0060】逆量子化部14は、可変長復号部13より入力された二次元データとなった、量子化された離散コサイン変換係数を、量子化幅、および、量子化行列に関する情報に基づいて逆量子化し、離散コサイン変換係数として加算器15に出力する。加算器15は、逆量子化部14より入力された離散コサイン変換係数から、動き補償誤差補正部20より入力される、離散コサイン変換係数の誤差成分を減算して、動き補償された離散コサイン変換係数を帯域制限部16に出力すると共に、動き補償誤差補正部20の加算器32に出力する。

【0061】帯域制限部16は、加算器15の出力として得られる動き補償処理された離散コサイン変換係数に基づいて、 8×8 のブロック毎に、水平方向の高域成分係数を削減し、量子化部17に出力する。

【0062】量子化部17は、符号量制御部23により入力される、目標とするビットレートに対応した量子化スケールコードに基づいて、帯域制限部16より入力された 8×8 の離散コサイン変換係数を量子化し、可変長符号化部18に出力すると共に、動き補償誤差補正部20の逆量子化部31に出力する。可変長符号化部18は、量子化部17より入力された量子化された離散コサイン変換係数をMPEG2方式で可変長符号化し、符号バッファ19に出力し、一時的に記憶させる。符号バッファ19は、低ビットレートに変換された画像圧縮情報を一時的に記憶して、出力すると共に、符号量制御部23に出力する。

【0063】正規化アクティビティ算出部22は、情報バッファ21よりビットレート、フレーム毎の発生ビット量、および、マクロブロック毎の量子化ステップサイズ、並びに、発生ビット量情報に基づいて、原画の輝度成分画素値より算出される正規化アクティビティを算出し、符号量制御部23に出力する。

【0064】符号量制御部23は、目標とするビットレートの、視覚特性に対応した量子化スケールコードを、正規化アクティビティ算出部22より入力された正規化アクティビティを利用して算出し、量子化部17に出力する。

【0065】動き補償誤差補正部20の逆量子化部31は、量子化部17より入力された量子化されている離散コサイン変換係数を、量子化幅と量子化行列に関する情報をもとに逆量子化し、加算器32に出力する。加算器32は、逆量子化部31より入力される離散コサイン変換係数と、逆量子化部15より入力される離散コサイン変換係数との差分を算出し、逆離散コサイン変換部33に出力する。逆離散コサイン変換部33は、この逆量子化部31より入力される離散コサイン変換係数と、逆量子化部15より入力される離散コサイン変換係数との差分値に逆離散コサイン変換処理を施し、動き補償誤差補

正情報（誤差データ）を生成して、ビデオメモリ34に出力し、格納される。

【0066】動き補償予測部35は、入力された画像圧縮情報（ビットストリーム）内における動き補償予測モード情報（フィールド動き補償予測モード、または、フレーム動き補償予測モード、および、前方向予測モード、後方向予測モード、または、双方向予測モード）、および、動きベクトル情報に基づいて、ビデオメモリ34内の誤差データを元に動き補償処理を実行し、空間領域での誤差補正值を生成して離散コサイン変換部36に出力する。離散コサイン変換部36は、動き補償予測部35より入力された空間領域での誤差補正值に離散コサイン変換処理を施すことで、周波数領域での誤差補正值を求めて、加算器15に出力する。

【0067】次に、図3のフローチャートを参照して、MPEG2方式で、かつ、20Mbpsのビットレートで圧縮された画像圧縮情報を、MPEG2方式で、かつ、16Mbpsのビットレートで圧縮された画像圧縮情報に変換する、ビットレート変換処理について説明する。ステップS11において、符号バッファ11は、入力された画像圧縮情報を一時的に記憶して、圧縮情報解析部12に出力する。ステップS12において、圧縮情報解析部12は、符号バッファより入力された画像圧縮情報（ビットストリーム）より、フレーム毎の発生ビット量、マクロブロック毎の発生ビット量、量子化スケールコード、量子化ステップサイズ、 q_scale_type フラグ、量子化行列、および、走査方式の情報を抽出して、情報バッファ21に格納させると共に、画像圧縮情報を可変長復号部13に出力する。

【0068】ステップS13において、正規化アクティビティ算出部22は、情報バッファ21に記憶された、入力となるマクロブロック毎の情報をを用いて以下に示す方法で原画の輝度成分画素値より算出された、正規化アクティビティを算出（逆算）する正規化アクティビティ算出処理を実行する。

【0069】ここで、図4のフローチャートを参照して、正規化アクティビティ算出処理を説明する。正規化アクティビティ算出部22は、入力となるMPEG2方式で圧縮された画像圧縮情報（ビットストリーム）は、量子化タイプを示す q_scale_type フラグがLinear Q、Non-Linear Qに限らず、以下の方法を用いて算出する。

【0070】ステップS31において、正規化アクティビティ算出部22は、 j 番目のマクロブロックの符号化に先立ち、入力ビットストリーム符号化時の仮想バッファの占有量を式(27)乃至式(29)により算出する。

【0071】

$$d_{ji-in} = d_{0i-in} + B_{j-in} - I - T_{i-in} \times (j-1) / MB \text{ cnt} \cdots (27)$$

$$d_{jp-in} = d_{0p-in} + B_{j-in} - I - T_{p-in} \times (j-1) / MB \text{ cnt} \cdots (28)$$

$$djb-in = d0b-in + Bj-in - 1 - Tb-in \times (j-1) / MB\ cnt \dots (29)$$

【0072】ここで、 $d0i-in$ 、 $d0p-in$ 、 $d0b-in$ は、入力画像圧縮情報（ビットストリーム）の符号化時を想定した仮想バッファの初期占有量を、 $Bj-in$ は、入力画像圧縮情報（ビットストリーム）のピクチャの先頭から j 番目のマクロブロックまでの発生ビット量を、 $MBcnt$ は1ピクチャ内のマクロブロック数をそれぞれ示す。 $Ti-in$ 、 $Tp-in$ 、 $Tb-in$ は、入力フレームの発生ビット量を示す。各ピクチャ符号化終了時の仮想バッファ占有量（ $dMBcnti-in$ 、 $dMBcntp-in$ 、 $dMBcntb-in$ ）は、それぞれ同一ピクチャタイプで、次のピクチャに対する仮想バッファ占有量の初期値（ $d0i-in$ 、 $d0p-in$ 、 $d0b-in$ ）として用いられる。

【0073】ステップS32において、正規化アクティビティ算出部22は、 j 番目のマクロブロックに対する量子化スケールコード $Qj-in$ を、以下のように示す式（30）により算出する。

$$Qj-in = dj-in \times 31 / r-in \dots (30)$$

【0074】 $Qj-in = dj-in \times 31 / r-in \dots (30)$
 【0075】 $r-in$ は、リアクションパラメータと呼ばれるフィードバックループの応答速度を制御するパラメータであり、入力ビットストリームの符号量（ビットレート）を用いて、以下のように示す式（31）で与えられる。

$$r-in = 2 \times \text{bit rate} / \text{picture rate} \dots (31)$$

【0077】尚、シーケンスの最初における仮想バッファの初期値は、以下のように示す式（32）乃至式（34）で与えられる。

$$d0i-in = 10 \times r-in / 31 \dots (32)$$

$$d0p-in = Kp \cdot d0i-in \dots (33)$$

$$d0b-in = Kb \cdot d0i-in \dots (34)$$

【0079】ここで、 Kp 、 Kb は、出力画像圧縮情報（ビットストリーム）の符号量制御に用いた方法と同様に、入力画像圧縮情報（ビットストリーム）に対して算出するものとする。

【0080】ステップS33において、正規化アクティビティ算出部22は、入力画像圧縮情報（ビットストリーム）の量子化ステップサイズ $Q_step_size-j-in$ を用いて式（35）により視覚特性を考慮した（視覚特性に対応した）量子化スケールコード $mquant-j-in$ を求める。

$$Q_step_size-j-in$$

$$mquant-j-in = Q_step_size-j-in / 2 \dots (35)$$

【0082】ステップS34において、正規化アクティビティ算出部22は、式（35）により求められた視覚特性を考慮した量子化スケールコード $mquant-j-in$ と、入力ビットストリーム符号化時の仮想バッファの占有量から算出される j 番目のマクロブロックに対する量子化スケールコード $Qj-in$ を用いて、入力画像圧縮情報（ビットストリーム）符号化時の符号量制御の際にフレーム内で用いられた正規化アクティビティ $Nact-j$ を式（3

6）により算出する。

$$Nact-j$$

$$Nact-j = mquant-j-in / Qj-in \dots (36)$$

【0084】このような処理により、従来では、離散コサイン変換係数により再量子化を行う際、マクロブロック毎の正規化アクティビティの算出のために、原画の画像圧縮情報を各画素毎の情報まで復元し、その輝度画素値の平均値と分散を求めていたが、その必要がなくなる。

【0085】ここで、図3のフローチャートの説明に戻る。

【0086】ステップS13において、正規化アクティビティ算出処理により正規化アクティビティが算出されると、ステップS14において、可変長復号部13は、入力された画像圧縮情報を、走査方式に基づいて、図5（A）で示すようなジグザグスキャン、または、図5（B）で示すようなオルタネートスキャンの順序で、量子化された離散コサイン変換係数を二次元データとして再配列し、量子化幅、および、量子化行列の情報と共に逆量子化部14に出力する。尚、図5（A）、図5（B）においては、 8×8 離散コサイン変換係数のスキャン順序が、数字により示されている。すなわち、図5（A）で示すようにジグザグスキャンでは、図中左上の「0」から順次、その右の「1」に、ついで、その左下の「2」に、さらに、その下の「3」、その右上の「4」、その右上の「5」、その右の「6」、その左下の「7」、その左下の「8」・・・のような順序で、最右列最下段の「63」までスキャンする。また、図5（B）で示すようにオルタネートスキャンでは、図中左上の「0」から順次、その下の「1」、さらにその下の「2」、さらにその下の「3」、右に1列、上に3段上の「4」、その下の「5」、右上の「6」、その下の「7」、その左下の「8」、その下の「9」、その左下の「10」・・・といった順序で、最右列最下段の「63」までスキャンする。

【0087】ステップS15において、逆量子化部14は、入力された量子化されている離散コサイン変換係数を、圧縮情報解析部12により抽出された、量子化スケールコード、および、量子化行列に基づいて逆量子化し、離散コサイン変換係数として加算器15に出力する。

【0088】ステップS16において、加算器15は、逆量子化部14より入力された離散コサイン変換係数から、動き補償誤差補正部20により後述する動き補償誤差補正処理された動き補償誤差を減算して、誤差を補正した離散コサイン変換係数を帯域制限部16に出力すると共に、加算器32に出力する。

【0089】ステップS17において、帯域制限部16は、加算器15より入力された誤差が補正されている離

散コサイン変換係数の水平方向高域成分係数を、ブロック毎に削減する。このとき、帯域制限部 16 は、輝度信号と色差信号とに分けて、水平方向高域成分係数を削減する。すなわち、輝度信号については、図 6 (A) で示すように、 8×8 の離散コサイン変換係数のうち、水平方向低域成分である 8×6 の離散コサイン変換係数（図中の黒丸）の値を保存し、残りをゼロ (0) に置き換える。色差信号に関しては、図 6 (B) に示すように、 8×8 の離散コサイン変換係数のうち、水平方向低域成分である 8×4 の離散コサイン変換係数（図中の黒丸）の値を保存し、残りをゼロ (0) に置き換える。入力となる画像圧縮情報（ビットストリーム）が、飛び越し走査画像のものである場合には、フレーム離散コサイン変換モードの際に、フィールド間の時間差に関する情報を、離散コサイン変換係数の垂直方向高域成分が含むことにより、その制限を行うことは大幅な画質劣化に繋がるため、垂直方向の帯域制限は行わない。また、この例に示したように、劣化がより人間の目に付きやすい輝度信号に比べ、より人間の目に付きにくい色差信号に対して、より大きく帯域制限を行うことで、画質劣化を最小限に抑えながら、再量子化の歪みを低減している。

【0090】尚、帯域制限部 16 における処理は、図 6 に示した方法以外であってもよく、例えば、ゼロ (0) と置き換える代わりに、予め用意した重み係数を離散コサイン変換係数の水平方向高域成分に乘じるようにして、同様の効果が生じるようにしてもよい。

【0091】ステップ S 18 において、符号量制御部 23 は、目標となるビットレート（今の場合 16Mbps）の画像圧縮情報に対応する量子化スケールコード $Qj\text{-out}$ を算出する。尚、この処理は、従来の方法における、式 (17) により演算されるものであり、その説明は省略する。

【0092】ステップ S 19 において、符号量制御部 23 は、ステップ S 18 の処理で算出した量子化スケールコード $Qj\text{-out}$ と、正規化アクティビティ $Nact\text{-}j$ に基づいて、以下の式 (37) を使用して、目標とするビットレートの、視覚特性を考慮した量子化スケールコード $mquant\text{-}j\text{-out}$ を求める。

$$\text{【0093】} \quad mquant\text{-}j\text{-out} = Qj\text{-out} \times Nact\text{-}j \cdots (37)$$

【0094】ステップ S 20 において、符号量制御部 23 は、ステップ S 19 の処理で演算された量子化スケールコード $mquant\text{-}j\text{-out}$ よりも、入力された画像圧縮情報の量子化スケールコード $mquant\text{-}j\text{-in}$ の方が小さいか否かを判定し、量子化スケールコード $mquant\text{-}j\text{-out}$ よりも、入力された元の画像圧縮情報の量子化スケールコード $mquant\text{-}j\text{-in}$ の方が大きいと判定した場合、その処理は、ステップ S 21 に進む。

【0095】ステップ S 21 において、符号量制御部 23 は、演算した量子化スケールコード $mquant\text{-}j\text{-out}$ を量

子化部 17 に出力する。

【0096】ステップ S 22 において、量子化部 17 は、符号量制御部 23 より入力された量子化スケールコード $mquant\text{-}j\text{-out}$ 、または、量子化スケールコード $mquant\text{-}j\text{-in}$ に基づいて、帯域制限部 16 より入力された、帯域制限されている画像圧縮情報を量子化して、可変長符号化部 18 に出力する。

【0097】ステップ S 23 において、可変長符号化部 18 は、入力された量子化されている離散コサイン変換係数のうち直流成分に関しては、1 ブロック前の直流成分係数を予測値としてその差分を可変長符号化し、その他の成分に関しては、予め設定された走査方式（ジグザグスキャン、または、オルタネートスキャン）に基づいて 1 次元の配列データに並べ替え、連続する 0 係数の数（ラン）、および、非 0 の係数（レベル）の組み合わせから可変長符号化を行い、符号バッファ 19 を介して入力された画像圧縮情報に比べて、低ビットレートの画像圧縮情報に変換する。

【0098】このとき、ブロック内の走査順で、それ以降の係数が 0 となった場合、EOB (End of Block) と呼ばれる符号を出力し、そのブロックに対する可変長符号化を終了する。例えば、入力となる画像圧縮情報（ビットストリーム）におけるブロックの係数が、図 7 (A) に示すようになっていた場合（図 7 中、黒丸は非 0 係数、白丸は 0 係数を示す）、図 6 (B) に示すような、離散コサイン変換係数の削減を施したとすると、非 0 係数の分布は図 7 (B) のようになる。ここで、図 5 (A) で示すようなジグザグスキャンにより、この離散コサイン変換係数が可変長符号化されると、最後の非 0 係数のスキャン番号は「50」に位置する係数（図 5 (A) の左から第 5 列目の上から第 7 段目の係数）になる。これに対して、走査変換を行い、図 5 (B) で示すようなオルタネートスキャンで、この離散コサイン変換係数が可変長符号化されると、最後の非 0 係数のスキャン番号は「44」に位置する係数（図 5 (B) の左から第 5 列目の上から第 7 段目の係数）となる。これにより、EOB 信号は、ジグザグスキャンの場合よりオルタネートスキャンにより検出される番号で設定されることになり、その分量子化幅としてより細かな値を割当ることができ、再量子化に伴う量子化歪を低減することが可能となる。

【0099】ステップ S 20 において、量子化スケールコード $mquant\text{-}j\text{-out}$ よりも、元の画像圧縮情報の量子化スケールコード $mquant\text{-}j\text{-in}$ の方が大きくないと判定した場合、ステップ S 24 において、符号量制御部 23 は、量子化スケールコード $mquant\text{-}j\text{-in}$ を量子化部 17 に出力する。

【0100】すなわち、ステップ S 20 の処理により、量子化スケールコード $mquant\text{-}j\text{-in} < \text{量子化スケールコード } mquant\text{-}j\text{-out}$ であると判定された場合、一度粗く量

量子化されたマクロブロックが再量子化されて、より細かく量子化されたことになる。粗く量子化されたことによる歪みは、細かく再量子化されることでは低減されない。うえ、このマクロブロックに対して、使用ビット量が多く使われることになるため、他のマクロブロックに割り当てられるビットの減少を招き、更なる画質劣化を引き起こすことになる。そこで、量子化スケールコード $mquant-j-in$ < 量子化スケールコード $mquant-j-out$ であると判定された場合、符号量制御部23は、量子化に使用する量子化スケールコードとして量子化スケールコード $mquant-j-out$ ではなく、量子化スケールコード $mquant-j-in$ を採用するようにしている。

【0101】次に、図8のフローチャートを参照して、動き補償誤差補正部20の動き補償誤差補正処理について説明する。

【0102】まず、動き補償誤差の生じる原因について述べる。原画像の画素値を0とし、この画素値0に対して、入力となる画像圧縮情報（ビットストリーム）の量子化幅Q1で復号した参照画素値L(Q1)と、再符号化後の画像圧縮情報（ビットストリーム）における、この画素値0に対する量子化幅Q2で復号された時の参照画像の画素値L(Q2)を考える。

【0103】Pピクチャ、または、Bピクチャのインターマクロブロックの画素の場合、差分値0-L(Q1)に離散コサイン変換が施されて符号化される。ところが、符号量（ビットレート）の削減が行われた、出力となる画像圧縮情報（ビットストリーム）は、0-L(Q2)を離散コサイン変換して符号化したものとみなして復号を行う。このとき、動き補償誤差補正部20による動き補償誤差補正処理が行われないものと仮定すると、一般的に、量子化幅Q1=量子化幅Q2ではないため、それぞれの差分値0-L(Q1)と差分値0-L(Q2)も異なることになる。このような現象が、Pピクチャ、Bピクチャで生じるため、動き補償にともなう誤差が発生する。

【0104】更に、Pピクチャで生じた画質劣化は、後続のPピクチャ、および、それを参照とするBピクチャに伝播し、更なる画質劣化を生じさせる。このような原理により、GOPの各ピクチャの復号が後段に向かって進むに従って、動き補償に伴う誤差の蓄積が原因で、画質が劣化し、次のGOPの先頭でまた良好な画質に戻るという現象（ドリフト）が生じることになる。そこで、動き補償誤差補正処理は、量子化幅Q1と量子化幅Q2が、相互に異なる値とならないように補償する処理である。

【0105】ステップS51において、逆量子化部31は、量子化部17より入力される離散コサイン変換係数を、情報バッファ21に記憶された量子化幅と量子化行列に関する情報をもとに逆量子化して、加算器32に出力する。ステップS52において、加算器32は、加算器15より入力される離散コサイン変換係数で、逆量子化部31より入力される離散コサイン変換係数を減算

し、その差分値を逆離散コサイン変換部33に出力する。ステップS53において、逆離散コサイン変換部33は、入力された差分値を逆離散コサイン変換し、その結果を動き補償誤差補正情報として、ビデオメモリ34に格納させる。

【0106】ステップS54において、動き補償予測部35は、入力となる画像圧縮情報（ビットストリーム）内における動き補償予測モード情報（フィールド動き補償予測モード、または、フレーム動き補償予測モード、および、前方向予測モード、後方向予測モード、または、双方向予測モード）、および、動きベクトル情報と、ビデオメモリ34内の誤差データに基づいて動き補償予測処理を施し、これによって生成された空間領域での誤差補正值を離散コサイン変換部36に出力する。ステップS55において、離散コサイン変換部36は、動き補償予測部35より入力された誤差補正值に離散コサイン変換処理を施し、周波数領域での誤差補正值として加算器15に出力する。

【0107】この逆離散コサイン変換部33、および、離散コサイン変換部36の処理については、例えば、

“A fast computational algorithm for the discrete cosine transform” (IEEE Trans. Commun., vol. 25, no. 9, pp. 1004-1009, 1977) で示されている様な高速アルゴリズムを適用するようにしても良い。また、逆離散コサイン変換部33、および、離散コサイン変換部36の処理について、水平方向の離散コサイン変換係数は、帯域制限部16において高域成分係数が0と置き直されているため、これに対する逆離散コサイン変換処理、および、離散コサイン変換処理を省くことで演算処理量を削減することができ、演算処理用のハードウェアの構成を簡素化することが可能となる。

【0108】さらに、画像における色差信号の劣化は、輝度信号の劣化に比べ、人間の目には分かり難いという特色を有しているため、上記の動き補償誤差補正処理を、輝度信号のみに適用することで、画質劣化を最小に保ちながら演算処理量を削減することができ、演算処理用のハードウェアの構成を簡素化することが可能となる。

【0109】また、Pピクチャにおける誤差はBピクチャに伝播するが、Bピクチャにおける誤差はそれ以上伝播しない。一方、Bピクチャには双方向予測モードを含み、膨大な演算処理量を必要とする。そこで、Pピクチャにのみ動き補償誤差補正処理を施すようにすることで、画質劣化を最小に保ちながら演算処理量を削減することができ、演算処理用のハードウェアの構成を簡素化することが可能となる。また、Bピクチャにおける処理が省略されることにより、ビデオメモリ34の容量を節約すること可能となる。

【0110】さらに、上述の例においては、誤差補正值の成分として8×8離散コサイン変換係数の全ての成分が

用いられているが、特に、離散コサイン変換モードがフレームDCTモードであり、入力となる画像圧縮情報（ビットストリーム）の走査方式が飛び越し走査である場合、動き補償誤差補正部20による動き補償誤差補正処理は、垂直方向高域成分の誤差を省略するといった処理により画質劣化を生じさせることになるが、水平方向に関しては高域成分となる4成分を省略しても、視覚的に影響するような画質劣化を生じないことが知られている。この事実を利用して、画質劣化を最小限に抑えながら、演算処理量を削減することができ、演算処理用のハードウェアの構成を簡素化することができ、さらに、ビデオメモリ34の容量を節約することが可能となる。

【0111】そこで、逆離散コサイン変換部33、および、離散コサイン変換部36においては、垂直方向には通常の8次の処理を施すが、水平方向に関しては、低域成分となる4次の係数のみを用いた処理を行うようにしてもよい。これにより、ビデオメモリ34の水平方向解像度を1/2とすることで、その容量を節約することが可能となる。

【0112】但し、この場合、動き補償予測部35は、1/4画素精度の動き補償処理が必要となる。この処理は、図9に示すように、画像圧縮情報（ビットストリーム）中の動きベクトルの値に応じて、線形内挿を行うことで動き補償誤差に伴う画質劣化を十分に抑制することが可能である。水平方向に対する処理としては、以下の手段が考えられる。

【0113】すなわち、図9（A）で示すように、元の動きベクトル mv 1乃至4の画素が存在する場合、図9（B）で示すように、水平方向に右方向に各画素間の1/4だけずれた位置の場合、動きベクトル mv 1'は、以下の式（38）に示すように計算される。

【0114】

$$mv\ 1' = 3/4 \times mv1 + 1/4 \times mv2 \cdots (38)$$

【0115】すなわち、それぞれの位置に対応する重み付けの平均により1/4画素ずれた位置の動きベクトルが計算される。同様にして、図9（C）で示すように、2/4（=1/2）画素分だけ右方向にずれた位置の場合、動きベクトル mv 1''は、以下の式（39）に示すように計算される。

$$f(0) = (F(0) + F(2)) \times A + F(1) \times C + (F(1) - F(3)) \times D \cdots (45)$$

$$f(1) = (F(0) - F(2)) \times A + F(3) \times B - (F(1) - F(3)) \times D \cdots (46)$$

$$f(2) = (F(0) - F(2)) \times A - F(3) \times B + (F(1) - F(3)) \times D \cdots (47)$$

$$f(3) = (F(0) + F(2)) \times A - F(1) \times C - (F(1) - F(3)) \times D \cdots (48)$$

【0124】離散コサイン変換を実行するときは、入力と出力が入れ替えられて、式（45）乃至式（48）演算が実行される。

【0125】更に、一般的に、輝度信号に比べ、色差信号は、その劣化が人間の目に付きにくいことが知られている。そこで、色差信号に関しては、逆離散コサイン変換33、および、離散コサイン変換部36において、上

【0116】

$$mv\ 1'' = 1/2 \times mv1 + 1/2 \times mv2 \cdots (39)$$

【0117】同様にして、図9（D）で示すように、3/4画素分だけ右方向にずれた位置の場合、動きベクトル mv 1'''は、以下の式（40）で示すように計算される。

【0118】

$$mv\ 1''' = 1/4 \times mv1 + 3/4 \times mv2 \cdots (40)$$

【0119】すなわち、上述の処理は、逆離散コサイン変換装置33においては、8次の離散コサイン係数のうち、低域4次係数のみに対して4次の逆離散コサイン変換を施し、離散コサイン変換部36においては、ビデオメモリ34から動き補償によって作られた画素領域での、各ブロック毎の8×8の誤差補正值に対して、水平方向には高域成分の4次の離散コサイン変換処理を施すことで4×8の周波数領域での誤差補正值が出力されるというものである。

【0120】また、水平成分の低域成分の4次の逆離散コサイン変換処理と、離散コサイン変換処理に、高速アルゴリズムを用いることで、更なる処理量の削減が可能となる。図10に、高速アルゴリズムの一例であるWangのアルゴリズム（Zhone de Wang, “Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform”, IEEE Tr. ASSP-32, No. 4, pp. 803-816, Aug. 1984）に基づいた手法を示す。図10において、 $F(0)$ 乃至 $F(3)$ を入力値とし、 $f(0)$ 乃至 $f(3)$ を出力値とする処理により、逆離散コサイン変換が実現され、 $f(0)$ 乃至 $f(3)$ を入力値とし、 $F(0)$ 乃至 $F(3)$ を出力値とすることで離散コサイン変換が実現される。ここで、演算子A乃至Dは、以下の式（41）乃至式（44）で示すように定義される。

$$【0121】A = 1/\sqrt{2} \cdots (41)$$

$$B = -\cos(\pi/8) + \cos(3\pi/8) \cdots (42)$$

$$C = \cos(\pi/8) + \cos(3\pi/8) \cdots (43)$$

$$D = \cos(3\pi/8) \cdots (44)$$

【0122】逆離散コサイン変換を実行するときは、以下のように示す式（45）乃至式（48）の演算が実行される。

【0123】

述の4×8の誤差補正信号のうち、色差信号の誤差補正成分に関しては、図11で示すように、垂直方向の低域成分の係数のみ（例えば4×4）を補正に用い、残りの高域成分に関しては0と置き換えることで、誤差補正に伴う演算処理量を更に削減することができ、更なる処理量の削減を行うようにしてもよい。ここで、黒丸は低域成分を示し、白丸は高域成分を示している。

【0126】以上によれば、画像圧縮情報のビットレートを変換する処理において、量子化スケールコードを演算する際、正規化アクティビティの演算に、式(22)乃至式(24)で示したような分散の演算を各画素毎にする必要がなくなるので、その演算処理量を低減させることができ、処理速度を向上させることが可能となる。

【0127】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行させることが可能な、例えば汎用のパーソナルコンピュータなどにプログラム格納媒体からインストールされる。

【0128】図12は、トランスコーダ1をソフトウェアにより実現する場合のパーソナルコンピュータの一実施の形態の構成を示している。パーソナルコンピュータのCPU101は、パーソナルコンピュータの全体の動作を制御する。また、CPU101は、バス104および入出力インタフェース105を介してユーザからキーボードやマウスなどからなる入力部106から指令が入力されると、それに対応してROM(Read Only Memory)102に格納されているプログラムを実行する。あるいはまた、CPU101は、ドライブ110に接続された磁気ディスク111、光ディスク112、光磁気ディスク113、または半導体メモリ114から読み出され、記憶部108にインストールされたプログラムを、RAM(Random Access Memory)103にロードして実行する。これにより、上述した画像処理装置1の機能が、ソフトウェアにより実現されている。さらに、CPU101は、通信部109を制御して、外部と通信し、データの授受を実行する。

【0129】プログラムが記録されているプログラム格納媒体は、図12に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク111(フレキシブルディスクを含む)、光ディスク112(CD-ROM(Compact Disc-Read Only Memory)、DVD(Digital Versatile Disc)を含む)、光磁気ディスク113(MD(Mini-Disc)を含む)、もしくは半導体メモリ114などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記録されているROM102や、記憶部108に含まれるハードディスクなどで構成される。

【0130】尚、本明細書において、プログラム格納媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理は、もちろん、必ずしも時系列的に処理されなくとも、並列的ある

いは個別に実行される処理を含むものである。

【0131】

【発明の効果】本発明の画像処理装置および方法、並びにプログラムによれば、第1のビットレートで符号化された画像データに基づいて、画像データを第1のビットレートで符号化するために必要な第1の量子化スケールコードを演算し、第1のビットレートで符号化された画像データに基づいて、第1の量子化スケールコードが視覚特性に応じて適応量子化された第2の量子化スケールコードを演算し、第1の量子化スケールコードと第2の量子化スケールコードに基づいて、正規化アクティビティを演算し、画像データを第2のビットレートで符号化するために必要とされる第3の量子化スケールコードを演算し、第3の量子化スケールコードと正規化アクティビティに基づいて、第3の量子化スケールコードが視覚特性に応じて適応量子化された第4の量子化スケールコードを演算するようにしたので、直交変換と動き補償により圧縮され、符号化された画像情報が、より低ビットレートの画像データに変換される処理を、高速に実現することが可能となる。

【図面の簡単な説明】

【図1】従来の量子化スケールコード決定処理を説明するフローチャートである。

【図2】本発明を適用したトランスコーダの一実施の形態の構成を示すブロック図である。

【図3】ビットレート変換処理を説明するフローチャートである。

【図4】正規化アクティビティ算出処理を説明するフローチャートである。

【図5】操作方式を説明する図である。

【図6】高域成分を削減する処理を説明する図である。

【図7】高域成分を削減する処理を説明する図である。

【図8】動き補償誤差補正処理を説明するフローチャートである。

【図9】1/4画素精精度の動き補償予測処理を説明する図である。

【図10】Wangの高速アルゴリズムによる逆離散コサイン変換処理と離散コサイン変換処理を説明する図である。

【図11】高域成分を削減する処理を説明する図である。

【図12】媒体を説明する図である。

【符号の説明】

1 トランスコーダ、11 符号バッファ、12 圧縮情報解析部、13 可変長復号部、14 逆量子化部、15 加算器、16 帯域制限部、17 量子化部、18 可変長符号化部、19 符号バッファ、20 動き補償誤差補正部、31 逆量子化部、32 加算器、33 逆離散コサイン変換部、34 ビデオメモリ、35 動き補償予測部、36 離散コサイン変換部

【図1】

【図11】

図1

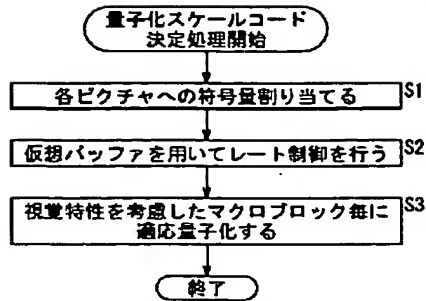
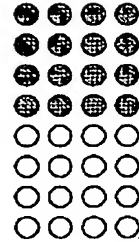


図11



【図2】

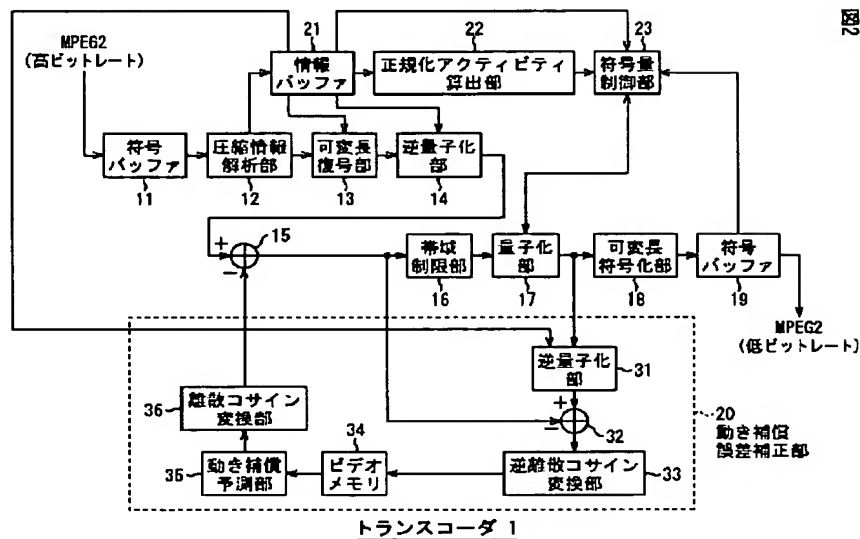


図2

【図4】

【図5】

図4

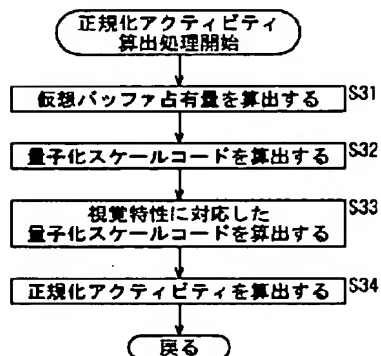


図5

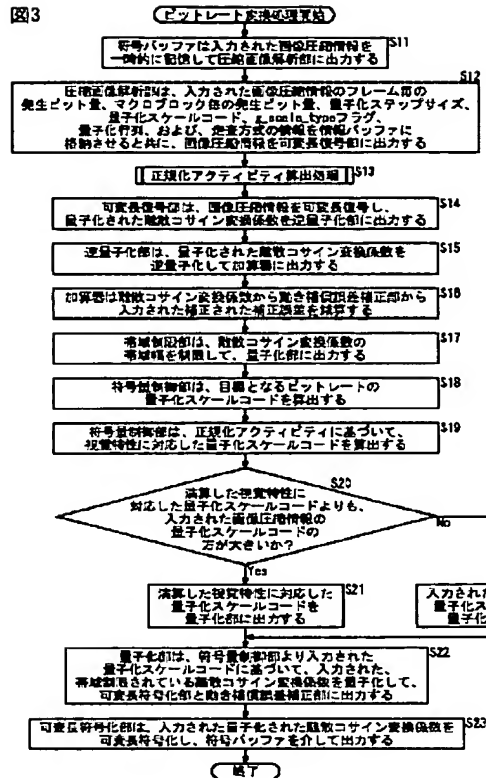
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	58	59	61
35	36	48	49	57	58	62	63

(A)

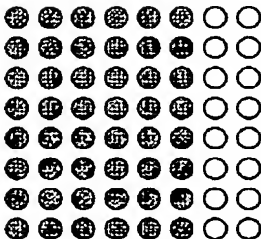
0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

(B)

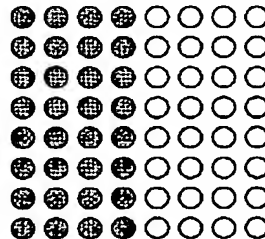
【図3】



【図6】

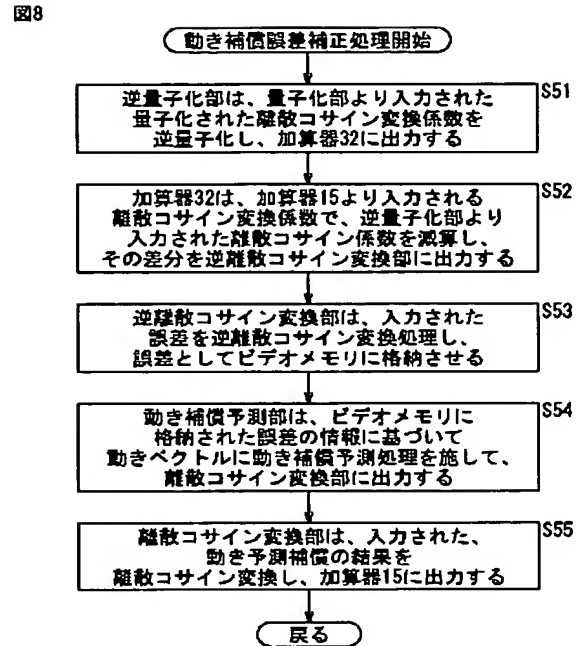


(A)



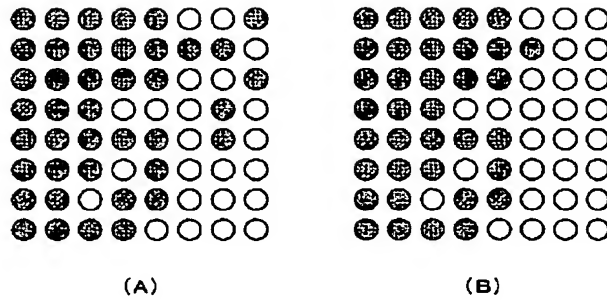
(B)

【図8】



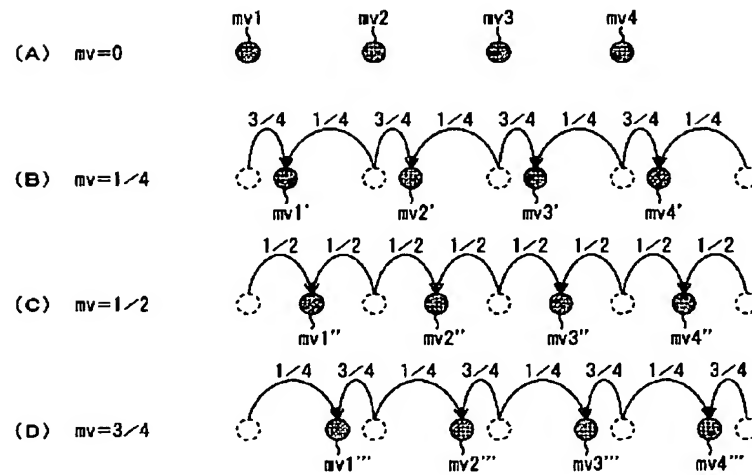
【図 7】

図 7



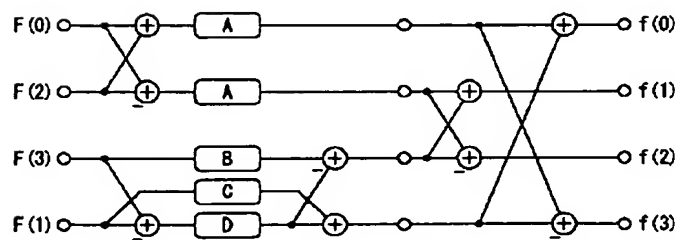
【図 9】

図 9



【図 10】

図 10



【図12】

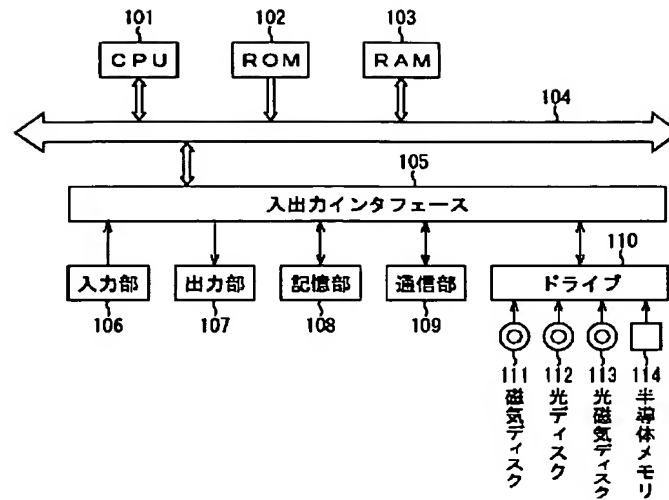


図12

フロントページの続き

(72) 発明者 矢ヶ崎 陽一
東京都品川区北品川6丁目7番35号 ソニ
株式会社内

Fターム(参考) 5C053 FA23 GA11 GB22 GB26 GB28
GB29 GB32 GB38 KA03
5C059 KK35 KK41 MA00 MA05 MA23
MC11 MC38 ME01 PP05 PP06
PP07 PP16 SS06 SS08 SS11
TA46 TA60 TB07 TC10 TD16
UA02 UA05 UA32 UA33
5J064 AA03 BA09 BA16 BB03 BC01
BC16 BC17 BC18 BC25 BD01